UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/659,642	09/10/2003	David G. Therrien	25452-015	3651

30623 7590 10/01/2008
MINTZ, LEVIN, COHN, FERRIS, GLOVSKY AND POPEO, P.C
ATTN: PATENT INTAKE CUSTOMER NO. 30623
ONE FINANCIAL CENTER
BOSTON, MA 02111

EXAMINER	
PHAM, MICHAEL	
ART UNIT	PAPER NUMBER
2167	
MAIL DATE	DELIVERY MODE
10/01/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/659,642	**Applicant(s)** THERRIEN ET AL.
	Examiner MICHAEL PHAM	**Art Unit** 2167

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 July 2008*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Claim Status

1. Claims 1-20 have been examined.

2. Claims 1-20 are pending.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 1 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. The claims recite in part "wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes" is not described in the specifications.

Claim Rejections - 35 USC § 101

5. The examiner interprets the term 'fileserver' as excluding transmission media, signals, or any form of energy, such that the claim clearly falls within a statutory class of invention as required under the terms of 35 U.S.C. 101.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-4, 8-12, and 16 rejected under 35 U.S.C. 103(a) as being unpatentable over US 20040049513 by Yakir et. al. (hereafter Yakir) further in view of US 20020055972 by Joseph Bernard Weinman JR (hereafter Weinman).

Claim 1 :

Yakir discloses the following claimed limitations:

"receiving, at a destination fileserver, a set of stub files associated with the set of files;"[figure 3 element 302.]

"initiating recovery of files in the set of files on the destination fileserver,"[0033, servers and SMS facilitate migration and remigration, and recall operations for files stored in storage environment 100. 0055, a recall is performed when a request is received to access a migrated file.]

"said initiating further comprises

using a stub file in the set of stub files, allowing access to a full content of a file associated with the stub file; and," [0046, stub file serves as an entity in the file system through which the original migrated file can be accessed. 0056, the original server may identify the repository server and the URI from information stored in the stub file to be recalled. Accordingly, using a stub file in the set of stub files (stub file), allowing access to a full content of a file (file) associated with the stub file (recall)]

"replacing each stub file with a full content of the file associated with the stub file; and"[0005, demigrates the requested data file from the repository storage location back to the original storage location. 0055, a recall is performed when a request is received to access a migrated file. Accordingly, replacing (recall) each stub file (stub file) with a full content of the file (file) associated with the stub file (stub file)]

"wherein said replacing includes

receiving a client request for a specified file in the set of files;" [0055, a recall is performed when a request is received to access a migrated file. Accordingly, receiving a client request (request) for a specified file in the set of files (file)]

"replacing the stub file associated with the specified file with a full content of the specified file; and," [0055, a recall is performed when a request is received to access a migrated file. Accordingly, replacing (recall) the stub file (stub file) associated with the specified file with a full content of the specified file (file)]

"replacing remaining stub files in the set of stub files with respective full contents of remaining files in the set of files."[0055, a recall is performed when a request is received to access a migrated file. Accordingly, replacing (recall) remaining stub files in the set of stub files (stub file) with respective full contents of remaining files in the set of files (file)]

Yakir does not explicitly disclose

"maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver;"

"initiating recovery of files in the set of files on the destination fileserver, wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes;"

On the other hand, Weinman discloses

"maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver;"[0040, and figures 2-5. Accordingly, maintaining, at the destination fileserver (server), a list of repository nodes that contain a replica of each file in the set of files (figures 3 and 4), and a list of files in the set of files stored at the destination fileserver (figure 5)]

"initiating recovery of files in the set of files on the destination fileserver, wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes;"[0015, and figures 3-5. Accordingly, initiating recovery of files in the set of files (create files) on the destination fileserver (server), wherein based on the list of files (figure 5) and the list of repository nodes stored (figure 3 and 4) at the destination fileserver (server), a replica of a file (copies) in the list of files (figure 5) is recovered from a repository node (figure 1) in the list of repository nodes (figure 3 and 4);]

Both Yakir and Weinman disclose methods of file storage and recovery, and are therefore within the same field of endeavor as applicant's invention. Yakir discloses migrating 0046 and 0055. Weinman discloses dynamically replicating the information at a number of sites and maintaing at least a predetermined minimum number of mirror sites containing the information, 0002. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied Weinman's disclosure above to the disclosure of Yakir for the purpose of backing up the files in case a disaster occurs. Doing so, would provide recovery for the migrated file in Yakir, in case the server results in accidental deletion, disaster that destroys data, and loss of location of which Weinman resolves (see 0051).

Claim 2 :

The combination of Yakir and Weinman discloses in Weinman "The method of claim 1 wherein the metadata is received at said the destination fileserver from a repository node in the list of repository nodes."[See figure 3. Accordingly, wherein the metadata (figure 3, capacity) is received at said the destination fileserver (server) from a repository node in the list of repository nodes (figure 3)]

Claim 3 :

The combination of Yakir and Weinman disclose in Yakir "selecting the destination fileserver for receiving the metadata and said the set of stub files."[0004, a stub file may contain attributes or metadata of the migrated file. 0009, moving a stub file.]

Claim 4 :

The combination of Yakir and Weinman disclose in Yakir "selecting a share of data for receiving

at said

destination fileserver."[0005, users and applications can access the migrated file as though the

file was

still stored in the original storage.]

Claim 8 :

Yakir discloses the following claimed limitations:

"a fileserver having:

a file system operative to store client files;"[figure 1 and 0004, a fileserver (figure 1) having: a file system operative to store client files (0004, filesystem)]

"a fileserver API operative to communicate with a repository;"[figure 1, a fileserver API (figure 1, SMS) operative to communicate with a repository (figure 1)]

"a fileserver file transfer module in communication with the file system and configured to transfer files for the file system to and/or from at least one repository; and"[See Figure 1 and 0004. Accordingly, a fileserver file transfer module in communication (figure 1, network) with the file system (figure 1, SMS) and configured to transfer files (0004, migrated) for the file system (file system) to and/or from at least one repository (figure 1)] "a recovery service in communication with the fileserver API and with the file system and configured to transfer a set of files," [0055, a recall is performed when a request is received to access a migrated file. Accordingly, a recovery service (0004, recall) in communication with the fileserver API (figure 1, sms) and with the file system (0004, file system) and configured to transfer a set of files (0004, migrated)]

"the recovery service having:

a receiving component configured to receive metadata and stub files associated with the set of files at the fileserver;"[Accordingly, a receiving component (figure 1)configured to receive metadata (0004, metadata) and stub files (0004, stub files) associated with the set of files (files)at the fileserver (figure 1)]

"a location updating component in communication with the receiving component and" [figure 4B element 452. Accordingly, a location updating component in communication with the receiving component (synchronize location)]

"wherein using a stub file in the set of stub files, said recovery service is configured to allow access to a full content of a file associated with the stub file; and"[0046, stub file serves as an entity in the file system through which the original migrated file can be accessed. 0056, the original server may identify the repository server and the URI from information stored in the stub file to be recalled. Accordingly, wherein using a stub file in the set of stub files (stub files), said recovery service (recalled) is configured to allow access to a full content of a file (file) associated with the stub file (stub file)]

> "a stub file replacement component in communication with the receiving component and configured to receive a client request for a specified file in the set of files," [Accordingly, a stub file (stub file) replacement component in communication with the receiving component (fig 3 and 4) and configured to receive a client request (request) for a specified file in the set of files (file)]
>
> "replace the stub file with the full content of the specified file associated with the stub file, and replace remaining stub files in the set of stub files with respective full contents of remaining files in the set of files."[0055, a recall is performed when a request is received to access a migrated file. Accordingly, replace the stub file (stub file) with the full content of the specified file (file) associated with the stub file (stub), and replace remaining stub files in the set of stub files (stub) with respective full contents of remaining files in the set of files (file)]

Yakir does not explicitly disclose:

> "configured to maintain a list of repository nodes that contain a replica of each file in the set of files and a list of files in the set of files stored at the destination fileserver;"
>
> "said recovery service is configured to initiate recovery of files in the set of files on the fileserver, wherein based on the list of files and the list of repository nodes stored at the fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes;"

On the other hand, Weinman discloses

"configured to maintain a list of repository nodes that contain a replica of each file in the set of files and a list of files in the set of files stored at the destination fileserver;"[0040, and figures 2-5. Accordingly configured to maintain a list of repository nodes that contain a replica of each file in the set of files (figure 3 and 4)and a list of files in the set of files stored at the destination fileserver (figure 5)]

"said recovery service is configured to initiate recovery of files in the set of files on the fileserver, wherein based on the list of files and the list of repository nodes stored at the fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes;"[0015, and figures 3-5. Accordingly, said recovery service is configured to initiate recovery of files in the set of files on the fileserver (create files), wherein based on the list of files (figure 5) and the list of repository nodes (figures 3 and 4) stored at the fileserver (server), a replica of a file in the list of files is recovered from a repository node in the list of repository nodes (disaster recovery);]

Both Yakir and Weinman disclose methods of file storage and recovery, and are therefore within the same field of endeavor as applicant's invention. Yakir discloses migrating 0046 and 0055. Weinman discloses dynamically replicating the information at a number of sites and maintaing at least a predetermiend minimum number of mirror sites containing the information. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied Weinman's disclosure above to the disclosure of Yakir for the purpose of backing up the files in case a disaster occurs. Doing so, would provide recovery for the migrated file in Yakir, in case the server results in accidental deletion, disaster that destroys data, and loss of location of which Weinman resolves (see 0051).

<u>Claim 9 :</u>

The combination of Yakir and Weinman disclose in Yakir:

"a policy cache operative to store a protection policy associated with a share;" [figure 1 element 114 a policy cache (figure 1) operative to store a protection policy associated with a share (figure 1 element 114, storage policies info)]

The combination of Yakir and Weinman disclose in Weinman

"a filter driver operative to intercept input/output activity initiated by client file requests and to maintain a list of modified and created files since a prior backup;"[0014-0015, when a request comes in from Miami, a new copy might be created in Maimi. Central server that keeps track of the global number of copies each object and their locations. Accordingly, a filter driver (method and means for data dispersion) operative to intercept input/output activity initiated by client file requests (request) and to maintain a list of modified and created files (figure 6) since a prior backup (copy)]

"a mirror service in communication with the filter driver and with the policy cache, the mirror service configured to prepare modified and created files in a share to be written to a repository as specified in the protection policy associated with the share."[0014, a copy might be created in Kansas from a version in Califorina. Accordingly, a mirror service (mirrored) in communication with the filter driver (data dispersion) and with the policy cache (server), the mirror service (mirror) configured to prepare modified and created files in a share (files) to be written to a repository (sites) as specified in the protection policy (consider such factors as the risk of disaster) associated with the share (data)]

Claim 10 :

The combination of Yakir and Weinman disclose in Weinman:

"a location cache in communication with the mirror service and configured to indicate which repository should receive an updated version of an existing file; and"[0014, a copy might be created in Kansas from a version in California. Accordingly, a location cache (california) in communication with the mirror service (mirror) and configured to indicate which repository (kansas) should receive an updated version of an existing file (copy)]

"a location manager coupled to the location cache and configured to update the location cache when the system writes a new file to a specific repository node."[0015, a central server that keeps track of the global number of copies of each object and their locations. Accordingly, a location manager (central) coupled to the location cache (locations) and configured to update the location cache when the system writes a new file to a specific repository node (keeps track of global number of copes of each object and locations)]

Claim 11 :

The combination of Yakir and Weinman disclose in Yakir:

"a local repository having:

a local repository node API configured to communicate with the fileserver API;" [figure 1, a fileserver API (figure 1, SMS) operative to communicate with a repository (figure 1)]

"a local repository file transfer module in communication with the fileserver file transfer module and configured to transfer files to the fileserver file transfer module; and"[figure 1, network]

"a data mover in communication with the local repository API and configured to supervise the replication of files from the local repository to the fileserver."[figure 1, SMS]

Claim 12 :

The combination of Yakir and Weinman disclose in Yakir:

"a remote repository having:

a remote repository node API configured to communicate with the network;"[figure 1]

"a remote repository file transfer module in communication with the local file transfer module and configured to transfer files to the fileserver file transfer module; and"[figure 1, network]

"a data mover in communication with the remote repository API and configured to supervise the replication of files from the remote repository to the fileserver."[figure 1, sms]

Claim 16 :

The combination of Yakir and Weinman disclose in Weinman "wherein the fileserver is configured to receives a metadata from a repository node in the list of repository nodes." [See figure 3.]

8. Claims 5-6, 15, and 18-19 rejected under 35 U.S.C. 103(a) as being unpatentable over US 20040049513 by Yakir et. al. (hereafter Yakir) further in view of US 20020055972

by Joseph Bernard Weinman JR (hereafter Weinman) and US 20020111929 by Pudipeddi et. al. (hereafter Pudipeddi).

Claim 5 :

The combination of Yakir and Weinman do not explicitly disclose:

"wherein the set of files is the set of files that have been accessed during a specified period; and"

"wherein the replacing each stub file step further comprises recursively replacing the stub files associated with the files that were accessed within the specified period until all the stub files associated with the set of files have been replaced."

On the other hand, Pudipeddi discloses "wherein the set of files is the set of files that have been accessed during a specified period; and" [0042, a storage management program on computer may target files on hard disk that have not been accessed for some predetermined period of time. Accordingly, wherein the set of files is the set of files (target files) that have been accessed (accessed) during a specified period (predetermined period of time)]

"wherein the replacing each stub file step further comprises recursively replacing the stub files associated with the files that were accessed within the specified period until all the stub files associated with the set of files have been replaced"[0042, a storage management program on computer may target files on hard disk that have not been accessed for some predetermined period of time. 0046, recall wherein the replacing each stub file (recall) step further comprises recursively replacing the stub files (recall) associated with the files (target files) that were accessed within the specified period (predetermined period of time) until all the stub files (stub) associated with the set of files have been replaced (recalled)]

Yakir, Weinman, and Pudipeddi are all within the same field of endeavor, namely, back up systems and restoration. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied the disclosure of Pudipeddi above to the combination of Yakir and Weinman, for the purpose of saving memory space in the file system and allowing for more than one file to be recalled at once. Doing so, would provide faster and more efficient recovery for the migrated file in Yakir, since Pudipeddi puts the requests for recalling data objects stored on tapes and removable disks in monotonically increasing sequences in order to minimize traversal of the backup mediums (0008).

Claim 6 :

The combination of Yakir, Weinman, and Pudipeddi disclose in Pudipeddi "wherein the specified period is a most-recent period" [0042, have not been accessed for some predetermined period of time (e.g. six months)].

Claim 15 :

The combination of Yakir, and Weinman do not explicitly disclose"wherein said replacing the sub file for the specified file is a higher priority task than replacing the stub files for non-requested files"

On the other hand, Pudipeddi discloses 0042, a stub that identifies the new location of each file may be retained on hardisk so that the file can be located later. 0048, a queue may be classified as either active or non active. 0006, a queue is active when its corresponding medium is mounted on a drive such that migrated files may be retrieved from the medium,; otherwise the queue is non-active. An active queue is processed until the queue is empty. Non-active queues wait until a drive becomes available and are then processed in the same manner Accordingly, wherein said replacing (recall) the stub file (stub) for the specified file (file) is a higher priority task (active) than replacing (recall) the stub files (stub) for non-requested files (non-active).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied Pudipeddi's disclosure above to the combination of Yakir and Weinman for the purpose of recalling data in an efficient manner thus reducing the back and forth traversals of the tape and reducing the wear and tear. Doing so, would provide faster and more efficient recovery for the migrated file in Yakir, since Pudipeddi puts the requests for recalling data objects stored on tapes and removable disks in monotonically increasing sequences in order to minimize traversal of the backup mediums (0008).

Claim 18 :

The combination of Yakir and Weinman do not explicitly disclose:

“wherein the set of files is the set of files that have been accessed during a specified period; and”

“wherein the recovery service is further configured to recursively replace the stub files associated with the files that were accessed within the specified period until all the stub files associated with the set of files have been replaced.”

On the other hand, Pudipeddi discloses "wherein the set of files is the set of files that have been accessed during a specified period; and" [0042, a storage management program on computer may target files on hard disk that have not been accessed for some predetermined period of time. Accordingly, wherein the set of files is the set of files (target files) that have been accessed (accessed) during a specified period (predetermined period of time)]
"wherein the recovery service is further configured to recursively replace the stub files associated with the files that were accessed within the specified period until all the stub files associated with the set of files have been replaced"[0042, a storage management program on computer may target files on hard disk that have not been accessed for some predetermined period of time. 0046, recall wherein the replacing each stub file (recall) step further comprises recursively replacing the stub files (recall) associated with the files (target files) that were accessed within the specified period (predetermined period of time) until all the stub files (stub) associated with the set of files have been replaced (recalled)]

Yakir, Weinman, and Pudipeddi are all within the same field of endeavor, namely, back up systems and restoration. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied the disclosure of Pudipeddi above to the combination of Yakir and Weinman, for the purpose of saving memory space in the file system and be able to allow the files to be recalled. Doing so, would provide faster and more efficient recovery for the migrated file in Yakir, since Pudipeddi puts the requests for recalling data objects stored on tapes and removable disks in monotonically increasing sequences in order to minimize traversal of the backup mediums (0008).

Claim 19 :

The combination of Yakir, Weinman, and Pudipeddi disclose in Pudipeddi “wherein the specified period is a most- recent period.” [0042, have not been accessed for some predetermined period of time (e.g. six months)]

9. Claims 7 and 17 rejected under 35 U.S.C. 103(a) as being unpatentable over US 20040049513 by Yakir et. al. (hereafter Yakir) further in view of US 20020055972 by Joseph Bernard Weinman JR (hereafter Weinman) and 5991753 by Michael J. Wilde (hereafter Wilde).

Claim 7 :

The combination of Yakir and Weinman disclose in Weinman wherein the metadata is associated with a file in the set of files (figure 5) and includes a fileserver name where the file was created (figure 3); a size of the file (figure 6); the list of all repository nodes that maintain a replica of the file (figure 5).

However the combination does not explicity disclose “a content checksum of the file when the file was first created or last modified.”

On the other hand, Wilde discloses col. 6 lines 4-5, normal attributes of files include their size, creation time, modification time, read/write/execute permissions and their owner. Accordingly, a content checksum of the file when the file was first created (creation time) or last modified (modification time).

Yakir, Weinman, and Wilde are all within the same field of endeavor, and further disclose systems in which back up and restore data. It would have been obvious to a person of an ordinary skill in the art to have provided Wilde's disclosure above to the combination of Yakir and Weinman for the purpose of being able to recognize a stub file, to locate the file, and preserve any altered attributes. In doing so, allows the system to identify which files should be and shouldn't be migrated more quickly, thereby improving Yakir's disclosure of merely migrating a file and replacing it with a stub file.

<u>Claim 17 :</u>

The combination of Yakir, and Weinman disclose in Weinman "wherein the metadata is associated with a file in the set of files" (figure 5) "and includes a fileserver name where the file was created" (figure 3); "a size of the file" (figure 6); "the list of all repository nodes that maintain a replica of the file" (figure 5).

The combination of Yakir, Weinman do not explicitly disclose "a content checksum of the file when the file was first created or last modified."

On the other hand, Wilde discloses col. 6 lines 4-5, normal attributes of files include their size, creation time, modification time, read/write/execute permissions and their owner. Accordingly, a content checksum of the file when the file was first created (creation time) or last modified (modification time).

Yakir, Weinman, and Wilde are all within the same field of endeavor, and further disclose systems in which back up and restore data. It would have been obvious to a person of an ordinary skill in the art to have provided Wilde's disclosure above to the combination of Yakir and Weinman for the purpose of being able to recognize a stub file, to locate the file, and preserve any altered attributes. In doing so, allows the system to identify which files should be and shouldn't be migrated more quickly, thereby improving Yakir's disclosure of merely migrating a file and replacing it with a stub file.

10. Claims 13-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 20040049513 by Yakir et. al. (hereafter Yakir) further in view of US 20020055972 by Joseph Bernard Weinman JR (hereafter Weinman) and US 20020069280 by Bolik et. al. (hereafter Bolik).

Claim 13 :

Yakir discloses the following claimed limitations:

"a file system operative to store client files;"[[figure 1 and 0004, file system]

"a policy component configured to store a protection policy associated with a set of files;"[0023, the information stored in database may include information related to storage policies and rules configured for the storage environment. Accordingly, a policy component (policies) configured to store a protection policy (storage policies and rules configured for storage environment) associated with a set of files (figure 1, files)]

"a fileserver file transfer module in communication with the file system and configured to transfer files for the file system to and/or from at least one repository; and,"[See Figure 1 and 0004. Accordingly, a fileserver file transfer module in communication (figure 1, network) with the file system (figure 1, SMS) and configured to transfer files (0004, migrated) for the file system (file system) to and/or from at least one repository (figure 1)]

"a location updating component"[figure 4b element 452, synchronizing location]

"said fileserver is configured to initiate recovery of files in the set of files on the fileserver,"[0033, servers and SMS facilitate migration and remigration, and recall operations for files stored in storage environment 100. 0055, a recall is performed when a request is received to access a migrated file. Accordingly, said fileserver is configured to initiate recovery of files in the set of files on the fileserver (recall)]

"wherein using a stub file in the set of stub files, said fileserver is configured to allow access to a full content of a file associated with the stub file by receiving a client request for a specified file in the set of files, replacing the stub file with the full content of the specified file associated with the stub file, and"[Accordingly, wherein using a stub file in the set of stub files (stub files), said fileserver is configured to allow access to a full content of a file (file) associated with the stub file (stub file) by receiving a client request (request) for a specified file in the set of files (file), replacing (demigrate) the stub file (stub file) with the full content of the specified file (file) associated with the stub file (stub file).]

"replacing remaining stub files in the set of stub files with respective full contents of remaining files in the set of files;" [See, 0055. Accordingly, replacing remaining stub files in the set of stub files (stub file) with respective full contents of remaining files in the set of files (file)]

Yakir does not explicitly disclose

"a mirror service in communication with the policy component, the mirror service operative to prepare modified and created files in a set of files to be written to a repository as specified in the protection policy associated with the set of files;"

"a fileserver API coupled to the mirror service and configured to communicate with a repository;"

"maintain a list of repository nodes that contain a replica of each file in the set of files and a list of files in the set of files stored at the destination fileserver"

"wherein based on the list of files and the list of repository nodes stored at said fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes;"

"determining a caching level for said fileserver; and"

"recursively, determining a utilization of the fileserver; comparing the caching level against the utilization; and"

"creating a file migration candidate list when the utilization exceeds the caching level;"

"staging out one candidate file;"

"replacing the candidate file with a stub file; and" "determining whether the utilization of the fileserver still exceeds the caching level."

On the other hand, Weinman discloses a mirror service (0009, mirror) in communication with the policy component (policies), the mirror service operative to prepare modified and created files (0029, data objects) in a set of files to be written to a repository (figure 1) as specified in the protection policy (policy) associated with the set of files (figure 1).

Weinman discloses a fileserver API (figure 1) coupled to the mirror service (0029, mirror) and configured to communicate with a repository (figure 1).

Weinman discloses maintain a list of repository nodes that contain a replica of each file in the set of files (figure 3)and a list of files in the set of files stored at the destination fileserver (figure 5)

Weinman discloses wherein based on the list of files the list of repository nodes (figure 3) stored at said fileserver (server), a replica of a file in the list of files (copy) is recovered from a repository node (figure 1) in the list of repository nodes (figure 3).

Both Yakir and Weinman disclose methods of file storage and recovery, and are therefore within the same field of endeavor as applicant's invention. Yakir discloses migrating 0046 and 0055. Weinman discloses dynamically replicating the information at a number of sites and maintaing at least a predetermiend minimum number of mirror sites containing the information. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied Weinman's disclosure above to the disclosure of Yakir for the purpose of backing up the files in case a disaster occurs. Doing so, would provide recovery for the migrated file in Yakir, in case the server results in accidental deletion, disaster that destroys data, and loss of location of which Weinman resolves (see 0051).

However, the combination of Yakir and Weinman do not explicitly disclose "determining a caching level for said fileserver; and"

"recursively, determining a utilization of the fileserver; comparing the caching level against the utilization; and"

"creating a file migration candidate list when the utilization exceeds the caching level;"

"staging out one candidate file;"

"replacing the candidate file with a stub file; and"

"determining whether the utilization of the fileserver still exceeds the caching level."

On the other hand, Bolik discloses the following:

"determining a caching level for said fileserver; and"[0070, once the file system exceeds a certain fill rate or runs out-of storage capacity the automigration process gets started. Accordingly, determining a caching level for said fileserver (fill rate/storage capacity)]

"recursively, determining a utilization of the fileserver; comparing the caching level against the utilization" [0070, once the file system exceeds a certain fill rate or runs out-of storage capacity the automigration process gets started. Accordingly, recursively, determining a utilization of the fileserver; comparing the caching level against the utilization (once the file system exceeds a certain fill rate or runs out-of storage capacity)].

“creating a file migration candidate list when the utilization exceeds the caching level” [0066, these candidates are merged into the existing list of candidates and then ranked for quality, thus incrementally improving the quality of migration candidates in the system. 0070, need for additional migration candidates. once the file system exceeds a certain fill rate or runs out-of storage capacity the automigration process gets started. Accordingly, creating a file migration candidate list (candidate list) when the utilization exceeds the caching level (once the file system exceeds)]

“staging out one candidate file;”[0070, consumes migration candidates from a dedicated automigration pool and signals the scout process to dump his set of migration candidates to disk or to transfer it into a migration queue via shared memory. Based on the newly dumped candidates list the automigration process can now start to migrate data to the remote HSM server. Accordingly, staging out one candidate file (migrate).]

”replacing the candidate file with a stub file; and”[0004, HSM is used for freeing up more expensive storage devices, typically magnetic disks, that are limited in size by migrating data files meeting certain criteria, such as the age of the file or the size to, to lower-cost storage media, such as tape, thus providing a virtually infinite storage space. To provide transparent access to all data files, regardless of their physical location, a stub file replaces the migrated file in the managed file system. Accordingly, replacing (replacing) the candidate file (migrated file) with a stub file (stub file)] "determining whether the utilization of the fileserver still exceeds the caching level."[0066, these candidates are merged into the existing list of candidates and then ranked for quality, thus incrementally improving the quality of migration candidates in the system. 0070, need for additional migration candidates. once the file system exceeds a certain fill rate or runs out-of storage capacity the automigration process gets started. 0048, A monitor daemon starts a master scout process and continuously monitors one or more file system. 0049, if the monitor daemon detects that the file system exceeded its threshold limits, it starts a master automigration process. Accordingly, determining whether the utilization of the fileserver still exceeds the caching level (continuously monitors).]

Yakir, Weinman, and Bolik are all within the same field of endeavor as applicant's invention, namely file storage systems and recovery. Bolik is resolves such problems as migrating data files quickly to minimize application latency, see 0010. Yakir discloses a system in order to move stub files to different locations and further migrate data in order to extend the storage. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied the disclosure of Bolik to the combination of Yakir and Weinman for the purpose of performing an automigration of files in order to save space on the fileservers. In doing so, the data migration in Yakir in order to save space on the server, is more efficiently done since Bolik's disclosure monitors the disk usage on the server and selects candidates ahead of time such that when a threshold is met, migration of the file is quickly decided.

Claim 14 :

The combination of Yakir, Weinman, and Bolik disclose “wherein said determining if the utilization of the fileserver still exceeds the caching level further comprises staging out another candidate file on the candidate list and again determining if the utilization of the fileserver exceeds the caching level.”[0066, these candidates are merged into the existing list of candidates and then ranked for quality, thus incrementally improving the quality of migration candidates in the system. 0070, need for additional migration candidates. once the file system exceeds a certain fill rate or runs out-of storage capacity the automigration process gets started. 0048, A monitor daemon starts a master scout process and continuously monitors one or more file system. 0049, if the monitor daemon detects that the file system exceeded its threshold limits, it starts a master automigration process.]

11. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over US 20040049513 by Yakir et. al. (hereafter Yakir) further in view of US 20020055972 by Joseph Bernard Weinman JR (hereafter Weinman), US 20020069280 by Bolik et. al. (hereafter Bolik), and US 20020111929 by Pudipeddi et. al. (hereafter Pudipeddi).

Claim 20 :

The combination of Yakir, Weinman, Bolik do not explicitly disclose:

“wherein the set of files is the set of files that have been accessed during a specified period; and”

“wherein the recovery service is further configured to recursively replace the stub files associated with the files that were accessed within the specified period until all the stub files associated with the set of files have been replaced."

On the other hand, Pudipeddi discloses "wherein the set of files is the set of files that have been accessed during a specified period; and" [0042, a storage management program on computer may target files on hard disk that have not been accessed for some predetermined period of time. Accordingly, wherein the set of files is the set of files (target files) that have been accessed (accessed) during a specified period (predetermined period of time)]
"wherein the recovery service is further configured to recursively replace the stub files associated with the files that were accessed within the specified period until all the stub files associated with the set of files have been replaced"[0042, a storage management program on computer may target files on hard disk that have not been accessed for some predetermined period of time. 0046, recall wherein the replacing each stub file (recall) step further comprises recursively replacing the stub files (recall) associated with the files (target files) that were accessed within the specified period (predetermined period of time) until all the stub files (stub) associated with the set of files have been replaced (recalled)]

Yakir, Weinman, and Pudipeddi are all within the same field of endeavor as applicant's invention, namely, storage systems and recovery. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied the disclosure of Pudipeddi above to the combination of Yakir and Weinman, for the purpose of saving memory space in the file system and be able to allow more than one file to be recalled at once. Doing so, would provide faster and more efficient recovery for the migrated file in Yakir, since Pudipeddi puts the requests for recalling data objects stored on tapes and removable disks in monotonically increasing sequences in order to minimize traversal of the backup mediums (0008).

Response to Arguments

12. Applicant's arguments with respect to claims 1-20 have been considered but are moot in view of the new ground(s) of rejection.

Applicant’s assert the following directed to the Prior cited references.

A. Page 13, Yakir requires both originating and destination fileservers are present in order to migrate files and is incapable of performing any data migration when originating file server is inactive. That this is in contrast to the present invention that provides disaster recovery operation when originating fileserver is non-operational, failed and/or non-existent.

In response, examiner respectfully disagrees. There is no need in Yakir for two fileservers. See figure 3. Furthermore, claim 1 is directed to transferring and recovery of files.

B. Page 14, Yakir fails to maintain a list of repository nodes that contain a replica of the file. Yakir simply replaces a requested stub file with its full content but does not continue to replace other non-requested files, which is contrary to the recitation of claim 1. Therefore, Yakir fails to disclose teach or suggest maintaining at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver, as recited in claim 1.

In regards to maintaining a list of repository nodes that contain a replica of the file and maintaining at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver. This is now moot in view of new grounds of rejection.

In regards to “Yakir simply replaces a requested stub file with its full content but does not continue to replace other non-requested files, which is contrary to the recitation of claim 1.” Applicant's argument that the references fail to show certain features of applicant’s invention, it is noted that the features upon which applicant relies (i.e., continue to replace other non-requested files) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Based on the specifications, the "continue to replace other non-requested files" appears to be disclosed as figure 11 element 68 of applicant’s specification. However, the claim only requires that remaining sub files be replaced. There is nothing regarding non-requested files. It would be obvious for the user to request remaining stub files to be replaced by merely running the method again.

C. Page 15, Wilde fails to disclose a list of repository nodes where replica of each file is stored, and therefore maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver. Page 15, Wilde fails to disclose initiating recovery of files in the set of files on the destination fileserver, wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes, as claimed in claim 1. Page 16, applicant asserts replacing remaining stub files in the set of stub files with respective full contents of remaining files in the set of files recited in claim 1 is not disclosed. Page 16, Wilde fails to disclose allowing access to a full content of a file associated with the stub file and replacing each stub file with a full content of the file associated with the stub file, wherein said replacing includes receiving a client request for a specified file in the set of files replacing the stub file associated with the specified file with a full content of the specified file, and replacing remaining files in the set of stub files with respective full contents of remaining stub files in the set of stub files with respective full contents of remaining files in the set of files as recited in claim 1.

In regards to a list of repository nodes where replica of each file is stored, and therefore maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver, this is moot in view of new grounds of rejection.

However in regards to the Wilde reference, Wilde does disclose initiating recovery of files in the set of files on the destination fileserver, since Wilde discloses resident and non-resident files. See for example col. 7 lines 30-49, which defines a resident and non-resident file. However, Wilde further discloses wherein based on the list of files as providing a candidate list, see col. 89 line 21. Wilde further discloses a replica of a file in the list of files is recovered from a repository node in the list of repository nodes as reloading a non-resident file, see col. 7 lines 49-65 and col. 8 lines 11-21. Wilde merely does not explicitly disclose a list of repository nodes stored at the destination server.

In regards to Wilde fails to disclose allowing access to a full content of a file associated with the stub file and replacing each stub file with a full content of the file associated with the stub file, wherein said replacing includes receiving a client request for a specified file in the set of files replacing the stub file associated with the specified file with a full content of the specified file, and replacing remaining files in the set of stub files with respective full contents of remaining stub files in the set of stub files with respective full contents of remaining files in the set of files as recited in claim 1. This arguement appears to be related to claim 1 above in which the remaining stub files are replaced even though they are non-requested. In response, Applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., continue to replace other non-requested files) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Based on the specifications, the "continue to replace other non-requested files" appears to be disclosed as figure 11 element 68 of applicant's specification. However, the claim only requires that remaining sub files be replaced. There is nothing regarding non-requested files. It would be obvious for the user to request remaining stub files to be replaced by merely running the method again.

D. page 17, Weinman does not maintain any list of nodes that have a replica of a file. Weinman is not concerned with maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver, as recited in claim 1.That therefore, Weinman fails to disclose initiating recovery of files in the set of files ion the destination fileserver wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes.

In response, the examiner respectfully disagrees. Please see new grounds of rejection as Weinman discloses "maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver;"[0040, and figures 2-5. Accordingly, maintaining, at the destination fileserver (server), a list of repository nodes that contain a replica of each file in the set of files (figures 3 and 4), and a list of files in the set of files stored at the destination fileserver (figure 5)] and "initiating recovery of files in the set of files on the destination fileserver, wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes;"[0015, and figures 3-5. Accordingly, initiating recovery of files in the set of files (create files) on the destination fileserver (server), wherein based on the list of files (figure 5) and the list of repository nodes stored (figure 3 and 4) at the destination fileserver (server), a replica of a file (copies) in the list of files (figure 5) is recovered from a repository node (figure 1) in the list of repository nodes (figure 3 and 4);].

E. Page 17-18, Lam does not cure the deficiencies of either one of the above references or their various combinations.

This is now moot in view of new rejections.

F. Page 18, Wilde, Weinman, and or Lam fail to resolve the deficiencies of Yakir. Even though all four references are in the same field of endeavor, all four references are deficient as they fail to disclose maintaining and initiating steps of the present invention. That there is no motivation to combine.

In regards to the initiating and maintaining steps, the examiner respectfully disagrees. Please see C above. In regards to the motivation, please see rejections. Furthermore, Applicant's field of endeavor is directed to primary data storage and comprehensive data protection. Wilde, Weinman, Lam, and Yakir are all related in that they are directed to data storage and systems that recover data. Applicant's further agree that these references are in the same field of endeavor. Therefore, a person of an ordinary skill in the art at the time the invention was made would have found that the references are analogous references and therefore combinable.

Lastly, in regards to the new combination, Both Yakir and Weinman disclose methods of file storage and recovery, and are therefore within the same field of endeavor as applicant's invention. Yakir discloses migrating 0046 and 0055. Weinman discloses dynamically replicating the information at a number of sites and maintaing at least a predetermined minimum number of mirror sites containing the information, 0002. It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to have applied Weinman's disclosure above to the disclosure of Yakir for the purpose of backing up the files in case a disaster occurs. Doing so, would provide recovery for the migrated file in Yakir, in case the server results in accidental deletion, disaster that destroys data, and loss of location of which Weinman resolves (see 0051).

Conclusion

13. The prior art made of record listed on pto-892 and not relied, if any, upon is considered pertinent to applicant's disclosure.

14. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to MICHAEL PHAM whose telephone number is (571)272-3924. The examiner can normally be reached on 9am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/M. P./
Examiner, Art Unit 2167

/Luke S. Wassum/
Primary Examiner, Art Unit 2167